
Трансформация Enterprise- и
Highload-приложений в HPC


HL
HighLoad++
Весна 2021



Типичный Enterprise

В организации	
Приложений	500-5000
CPU-ядер	1-5 млн
Хранение	10-100 PB
Дата-центров	5-20
Стоимость оборудования	1-5 млрд. USD
Количество поддержки	300-1500 инженеров поддержки и технических контактов в приложениях
Стоимость функционирования	150-500 млн. USD / год

Типичный Enterprise - история

- 50–100 лет истории бизнеса и 40+ лет ИТ
- Использование всех технологий когда-либо существовавших
- IBM 360, мейнфреймы, AS/400, AIX, HPUX, Sparc/Solaris...
- Kobol, C/C++ 20-летней давности...

Типичный Enterprise – история

- Ключевой фактор инфраструктуры — надежность

- Типичный дизайн — пара дата-центров:
    - В основном ДЦ — active/passive HA
    - Во вспомогательном ДЦ — DR, тестовые и разработческие среды
    - Многократное дублирование сетевых устройств

Типичный Enterprise – история

- Ключевой фактор разработки — быстрая реализация бизнес-задач

- И даже разработка отдельных платформ для конкретных трейдеров/групп или продуктов

- Почему — только новые продукты приносят деньги.
 Как только продукт появляется у конкурентов, цена падает до порога прибыльности

Типичный Enterprise – последствия

- 40 лет гонки за бизнес-задачами породили колоссальную сложность управления софтом и инфраструктурой
- Оптимизацией с точки зрения стоимости владения никто не занимался
- Оптимизацией с точки зрения стоимости инфраструктуры никто не занимался

Типичный Enterprise – последствия

Сложность достигла такого уровня, что стоимость владения начала существенно влиять на стоимость ведения бизнеса, даже с учетом достаточной прибыльности инвестиционных банков

Колоссальная инертность — «40 лет всё было хорошо, почему проблемы возникли сейчас?»

Типичный Enterprise – последствия

- Средняя загрузка CPU 5-15%
- Средняя загрузка сети 1-5-10%
- Двадцать дата-центров полностью забиты оборудованием

Как следствие:

Очень длинное время доставки оборудования	Очень много старого оборудования
Сложная 3-4 уровневая топология сети	Проблемы с расширяемостью
Плохая скорость сети	Плохая скорость систем хранения

Типичный Enterprise – последствия

Только когда проблемы ИТ стали серьезно мешать бизнесу, бизнес принял решение о трансформации ИТ и стал интересоваться, что и как можно улучшить

Вот так могу

Oak Ridge Supercomputer Center (суперкомпьютер Summit)		
Приложений	500-10000	**<100**
CPU-ядер	1-5 млн	**2.4 млн* (вместе с GPU)**
Хранение	10-100 PB	**250 PB**
Дата-центров	5-20	**600м2**
Стоимость оборудования	1-5 млрд. USD	**0.25 млрд. USD**
Количество поддержки	300-1500 инженеров	**30 инженеров**
Run cost	150-500M USD / год	**50M USD/ год**

Oak Ridge Supercomputer Center (суперкомпьютер Summit)

30 инженеров – никакой сложности поддержки

Нет диверсификации оборудования

Малая полная стоимость оборудования

Малое количество единиц оборудования

Нет проблем с расширением

Гипер-скоростная и простая сеть

Гипер-скоростная система хранения

Что дает HPC-оборудование – сеть



Mellanox CS8500

800 портов по 200Гб/сек
320 Тб/сек полная скорость переключений
1-2х уровневая сеть
130нс латентность

Что дает HPC-оборудование – хранение



Supermicro NVMe-стойка

672M операций в секунду
840GB/s пропускная способность
10.7 PB емкость (8TB NVMe SSD)

Что дает HPC-оборудование – сервера



Atos BullSequana XH2000

32 лезвия

3 сервера на лезвие

В стойке:

- 12288 ядер (Epyc 64 ядра)
- 196 ТБ памяти (128GB модуль)
- 19.2Тб/sec пропускной способности сети

Как использовать HPC-оборудование в Enterprise/Highload?

Лучше писать Software ☺

Как лучше писать Software?

Отрицание

Гнев

Торг

Депрессия

Принятие

Микросервисы в HPC-стиле

Площадь ячейки равна объему работы

Микросервисы

Микросервисы в HPC-стиле

Площадь ячейки равна объему работы

Автоматическое масштабирование	Разделение процессов сохранения данных и обработки
Асинхронность & пакетная обработка	Низко-латентные высоко-скоростные протоколы (UDP)
Фреймворки контроля бизнес/технических показателей и среды	
Полный мониторинг платформ	

Автоматическое масштабирование

Масштабирование – увеличение/уменьшение количества реплик/инстансов в зависимости от:

1. Простой вариант – в зависимости от загрузки CPU/памяти/сети
2. Сложный – в зависимости от бизнес-процессов

Разделение процессов сохранения данных и обработки

- Быстрые операции обработки данных в памяти не должны упираться в конце в ожидание записи в базу данных или другое персистентное хранилище

Варианты:

1. Запись результатов в распределенный кэш
2. Оправка сообщением

Асинхронность & пакетная обработка

Количество операций в распределенных системах ограничено задержками.

При задержках 1мс, возможно только 1000 операций в секунду

Правило:

- Одна операция должна работать с десятками и сотнями объектов, а не с одним объектом – асинхронность с клиентом + пакетная обработка

Низко-латентные высоко-скоростные протоколы (UDP)

Протоколы с подтверждением в сети IB не нужны

Приоритет – нано-секундная латентность

Варианты библиотек UDP-протокола:

1. smUDP (Одноклассники)
2. Aeron (Martin Thompson)

Фреймворки контроля бизнес/технических показателей и среды

Борьба с перегрузками и OOM

1. Ограничение количества запросов
2. Ограничение по использованию памяти/цпу
3. Проверки связности с другими сервисами (trottling)

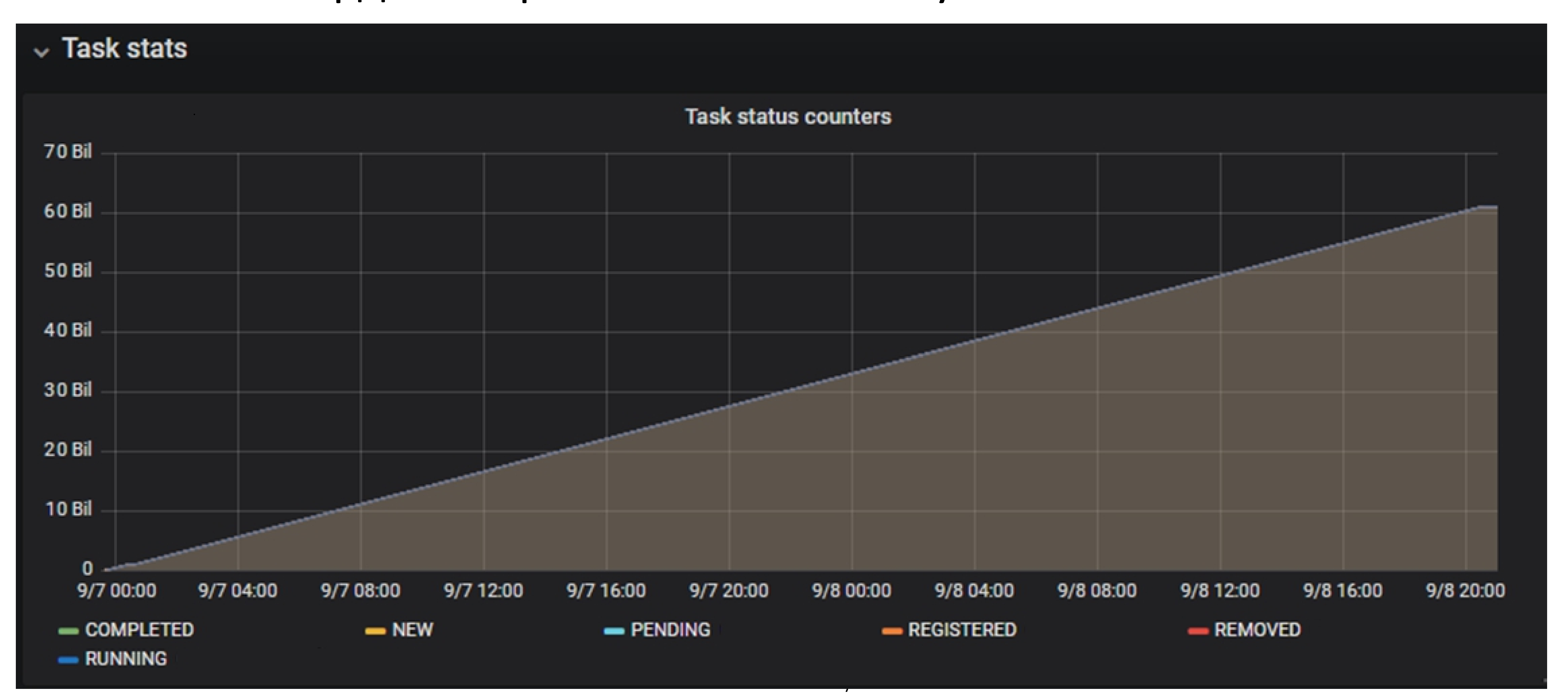
Полный мониторинг платформ

- Мониторинг индивидуальных инстансов
- Мониторинг внутренних параметров (GC, cpu usage/ram usage/thread usage/ времена get/put на map/queue/list/set)
- Мониторинг бизнес-показателей (операций в секунду/продолжительность операций)

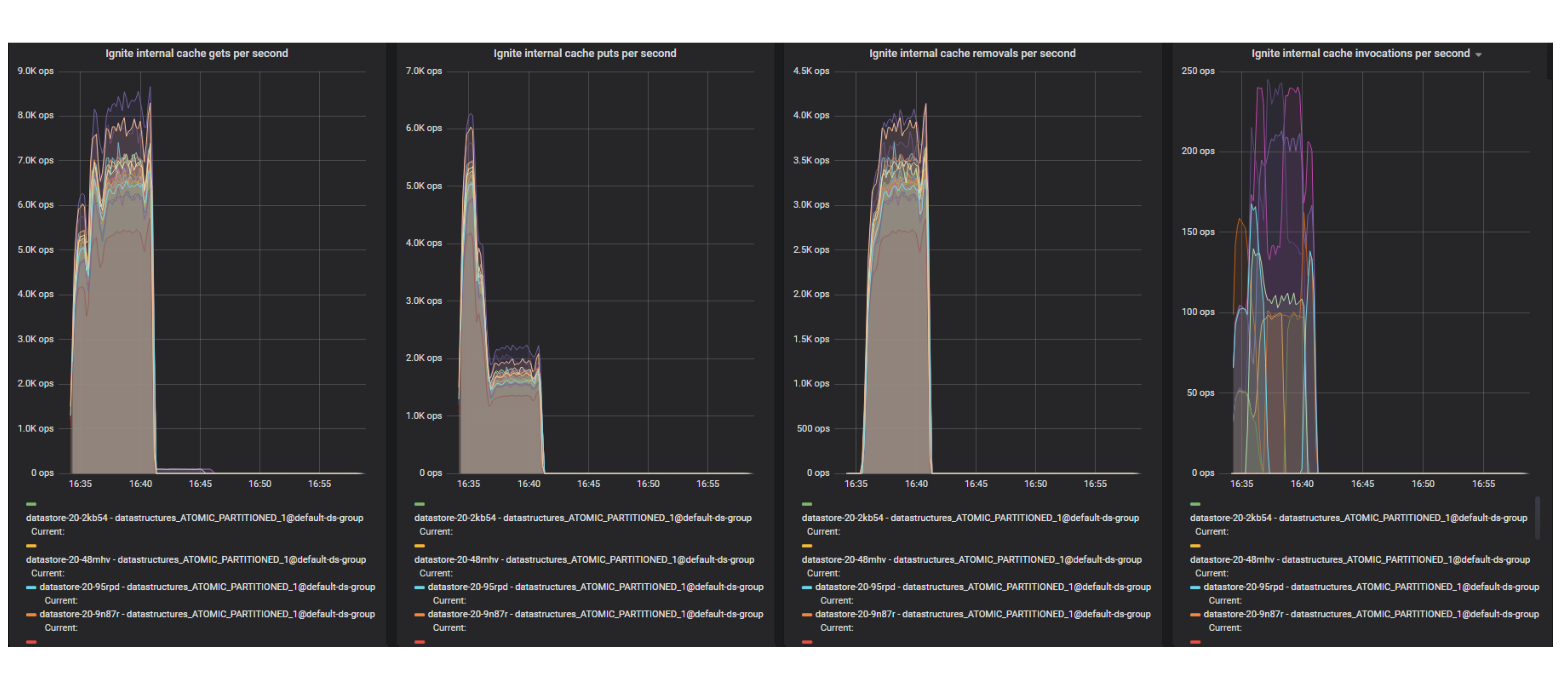
- Забудьте о логах – только метрики в Prometheus
- Логи только для неструктурированных ошибок

Что же получим в результате?

60 миллиардов запросов за 44 часа – Гугл на обычном железе



Мониторинг низкоуровневых операций



Мониторинг асинхронности (thread pool)

Здесь будет слайд с 85К распределенных операций в секунды, как безопасники согласуют

Q&A